2013-09

# Rapid network design

Garcia, Timmy J.

Monterey, California: Naval Postgraduate School


# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**RAPID NETWORK DESIGN**

by

Timmy J. Garcia

September 2013

| | |
|---|---|
| Thesis Advisor: | Geoffrey G. Xie |
| Second Reader: | Thomas Otani |

**Approved for public release; distribution is unlimited**

THIS PAGE INTENTIONALLY LEFT BLANK

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704–0188*



| 1. REPORT DATE *(DD–MM–YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From — To)* |
|---|---|---|
| 23–9–2013 | Master's Thesis | 2102-06-01—2104-10-31 |

**4. TITLE AND SUBTITLE**

RAPID NETWORK DESIGN

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Timmy J. Garcia

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Naval Postgraduate School
Monterey, CA 93943

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Department of the Navy

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

Approved for public release; distribution is unlimited

**13. SUPPLEMENTARY NOTES**

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.IRB Protocol Number: N/A

**14. ABSTRACT**

Network planning is a key element in the Marine Corps' communications planning process. The ability to design and provide reliable network architecture directly affects the commander's ability to control operations in an operational environment. Command-and-control systems technologies continue to change and evolve, adding complexity to network design. Portions of the current process of designing packet-switched networks are extremely prone to human design faults, which can adversely affect the reliability of the network. This thesis proposes an application prototype for network design that automates the creation of network configuration files. It describes the benefits achievable for development of such an application. Lastly, we demonstrate a working prototype that successfully produced configurations files that can easily be uploaded to network devices and create a functioning packet-switch network.

**15. SUBJECT TERMS**

network design, network topology, packet-switching networks, routing protocols, data communications, network communications

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | 19b. TELEPHONE NUMBER *(include area code)* |
| Unclassified | Unclassified | Unclassified | UU | 75 | |

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release; distribution is unlimited**

**RAPID NETWORK DESIGN**

Timmy J. Garcia
Captain, United States Marine Corps
B.B.A, University of Oklahoma, 2006

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

from the

**NAVAL POSTGRADUATE SCHOOL**
**September 2013**

Author: Timmy J. Garcia

Approved by: Geoffrey G. Xie
Thesis Advisor

Thomas Otani
Second Reader

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Network planning is a key element in the Marine Corps' communications planning process. The ability to design and provide reliable network architecture directly affects the commander's ability to control operations in an operational environment. Command-and-control systems technologies continue to change and evolve, adding complexity to network design. Portions of the current process of designing packet-switched networks are extremely prone to human design faults, which can adversely affect the reliability of the network. This thesis proposes an application prototype for network design that automates the creation of network configuration files. It describes the benefits achievable for development of such an application. Lastly, we demonstrate a working prototype that successfully produced configurations files that can easily be uploaded to network devices and create a functioning packet-switch network.

THIS PAGE INTENTIONALLY LEFT BLANK

# Table of Contents

# List of Figures

# List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

# List of Acronyms and Abbreviations

**ACE** aviation combat element
**ACL** access control list
**ARP** address resolution protocol
**CE** command element
**COA** course of action
**C2** command and control
**GCE** ground combat element
**LCE** logistics combat element
**MAGTF** Marine Air Ground Task Force
**MWCS** Marine Wing Communications Squadron
**NPS** Naval Postgraduate School
**POTS** plain old telephone system
**PSN** packet-switched network
**STIG** security technical implementation guides
**VLAN** virtual local area network
**VTC** video telephone conference
**WAN** wide area network

THIS PAGE INTENTIONALLY LEFT BLANK

# Acknowledgements

First and foremost, I would like to thank my loving wife and our rambunctious son for all their support the past two years. My wife's unwavering devotion and understanding allowed me to successfully navigate through this program and complete my studies. Words cannot describe my gratitude to both my wife and son for everything they do and continue to do to support me.

I would like to express my gratitude to both thesis advisors, Dr. Geoffrey G. Xie and Dr. Thomas Otani. Your expertise and sound guidance helped me navigate and successfully complete the thesis process. Your technical insight and encouragement helped make this thesis possible. Thank you!

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 1:
# Introduction

Communications networks are a vital capability of military commanders. Military communications networks provide a commander with the ability to quickly build situational awareness of an operation or training exercise and exert proper command and control. Command and control (C2) systems have progressively moved away from a primary reliance on single-channel radios, and innovation has led to the development of advanced communications systems. These systems have provided commanders with additional capabilities while also adding additional design complexities. These new communications systems increasingly rely on packet-switched networks (PSN) and place increased demand on the C2 networks. For instance, the plain old telephone system (POTS) nodes are typically interconnected via copper cabling and were separate from the packet-switched network. The interconnected POTS nodes are referred to as a circuit-switched network. POTS nodes can now be interconnected over the packet-switched network, which increases redundancy instead of relying solely on a single path. This allows each node to share its calling database across the PSN, which can sometimes be quite large in size.

From our experience and observations new communications systems fielded in the Marine Corps, are designed with a capability to connect to PSN. Even though C2 systems technologies have increased network design complexity, the military network design process has changed little, making it difficult to accommodate these new demands. C2 depends on a predictable and reliable PSN, which is driven by correct and consistent network configuration on all network devices. The majority of network engineers and operators still configure network devices in a manual and adhoc fashion. Manual configuration is highly prone to user error, that could lead to unscheduled network outages, degraded performance, or increase security vulnerabilities [1]. According to a recent study, design faults and mis-configured devices account for over half of network outages [2], and create additional vulnerabilities in the network.

The Marine Corps' current network design process has proven effective to meet mission objectives, yet has been subjected to significant design flaws and misconfigurations that have significantly affected the networks' resiliency and security. PSNs are heavily relied upon by commanders, yet the networks are typically implemented in an adhoc fashion. Mis-configurations are not uncommon as network operators implement even a single network design. The scope of

this thesis is to take the first steps toward eliminating these errors by automating the process of network design.

## 1.1 Goals

Using previous work performed in this area and an understanding of the Marine Corps process for network design, this research will develop a software prototype for the Marine Corps that provides an automated network design solution. The goals of this thesis are to:

- Determine the current tactics, techniques, and procedures utilized for network design in the Marine Corps and identify which processes can be easily automated. Two of the design steps that are candidates for automation are discussed in Chapter 2.
- Determine the current set of software tools used by network planners to create a software solution that performs equivalent functions.
- Develop an automated software proof-of-concept solution that is capable of automating network configuration.
- Determine if the tool developed here can replicate a standard Marine Corps network design and automatically produce network device configuration files that can be loaded onto physical network devices.

## 1.2 Scope

The principle fighting unit of the Marine Corps is the Marine Air Ground Task Force (MAGTF). The MAGTF consists of four principle elements: command element (CE), ground combat element (GCE), aviation combat element (ACE), and logistics combat element (LCE). Each element possesses a communications unit, that is responsible for the network communication. The network design and implementation process is consistent across all these elements. We have a firm understanding and working knowledge of planning, installing, operating, and maintaining communications networks in the ACE. Our primary focus for this thesis is ACE network design and replication of their mission requirements for network support. The unit responsible for networking services for the ACE is Marine Wing Communications Squadron (MWCS). The mission of the communications squadron is to support three communications nodes, one primary site and two secondary sites [3]. C2 networks employed by MWCS consist of a mix of PSN and CSN networks. The PSN network follows a rigid hierarchical design pattern consisting of a core routing layer, a distribution layer, and access layer. We explain the hierarchical design pattern further in Chapter 2.

Figure 1.1 and Figure 1.2 are similar template diagrams that are used in the development and employment of Marine Corps communications networks. Figure 1.1 depicts three separate sites that are connected via a routing backbone. Figure 1.2 shows the switching architecture of the main site the was depicted in Figure 1.1. The other two sites would have a similar switching architecture.

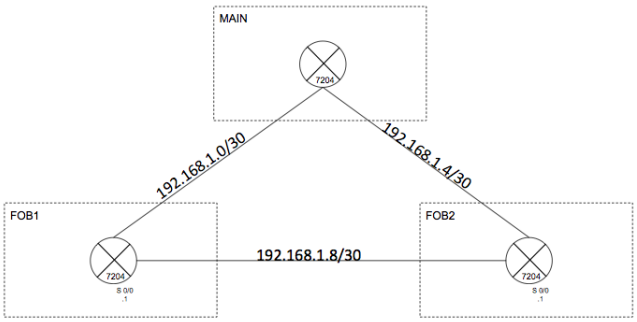


Figure 1.1: USMC Routing Diagram

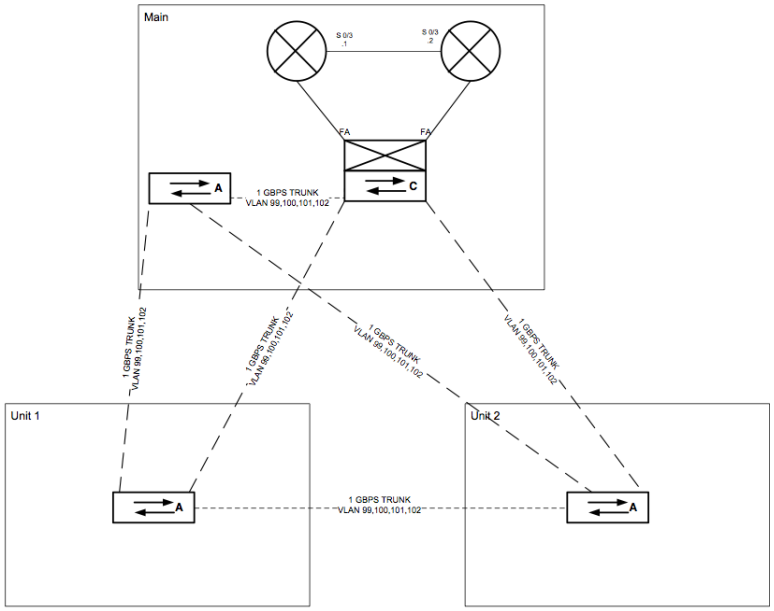


Figure 1.2: USMC Switch Diagram

The objective of this work is to automatically replicate network configuration for the core routing and switching architectures. The thesis will focus on VLAN design and access control list (ACL) placement utilizing the automated framework currently available for these design elements. To demonstrate feasibility of such automation the proof-of-concept will demonstrate the correctness of the algorithms by implementing data services such as video telephone conference (VTC) and exchange. Understanding that there are many aspects that are incorporated in network design, this solution is the first step toward automation and will only address a small subset of the process. The final proof-of-concept will demonstrate the feasibility of automating network design and produce correct network device configuration files.

## 1.3 Results

By creating a working application prototype for automated network design and applying the use of systematic algorithms, we are able to provide the following contributions through the use of RND:

- Developed an application that provides similar functionality and capabilities of current design software, while integrating a systematic approach to network design.
- Demonstrated the ability to accurately automate two elements of the network design process: VLAN router and route-selection and ACL placement.
- Demonstrated the ability to synthesis network engineers' logical network diagram and produce accurate network device configuration files.
- Offered recommendations for future work.

This tool represents a step toward automating clean-slate network design. With additional research and further development of the application to meet Marine Corps specifications, RND may provide a viable means to help reduce network design and implementation error and increase network design proficiency.

## 1.4 Organization

The rest of the thesis is organized as follows:

Chapter 2 provides background information to tie together the scope and the context of the problem. It discusses research on systematic network design followed by design techniques utilized by the United States Marine Corps. Chapter 3 describes the algorithms in detail from state-of-the-art methods presented in [4]. Then discussed are additional algorithms developed

to compute and create network configuration files for a given network design. Chapter 4 provides an overview of the user interface developed for the RND application. Chapter 5 provides an overview of the methodology used to test the application and the results of implementing systematic network design techniques. Chapter 6 provides a summary of findings and outlines future work to enhance the application developed in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 2:
# Background

## 2.1 Network Design

Computer networks have become integral to the way we live, communicate, conduct business, and entertain ourselves. We expect our email at a moment's notice and our online transactions to complete without worries, yet we could care less how these computer networks were built or designed. We just want them to work. Network designers and operators, however, are intimately involved in the understanding of the planning, installation, operation, and maintenance (PIO&M) of these networks.

Each step of the PIO&M process has its own nuances that make them particularly challenging. The planning phase involves two interrelated tasks: analyzing customer requirements and developing a network design capable of supporting given requirements. Customer requirements analysis involves network designers defining network traffic patterns, application usage, number of users, security, and a variety of other elements that help define the network's design. Customer requirements typically involve such items as, the network must support video-telephone conferencing and 100+ users with email access, while maintaining 90 percent reliability. Gaining a thorough understanding of the customer requirements and translating them into easily understood design tasks makes this step particularly difficult.

Once customer requirements have been defined, network design can begin. Network design synthesizes customer requirements and enables the network engineer to create a logical representation of the physical network. The logical network representation is an abstraction of the physical network and depicts logical connections between network devices, illustrating which physical devices can share information. It does not necessarily represent physical connections or locations. A thorough analysis of customer requirements drives the network design process to develop a physical packet-switched network that supports operational goals for the customer. This may seem like a simple task on the surface; however, with increasing reliance on interconnectivity by customers and challenging complexity, security, performance requirements, the design process becomes a very complex task. There are several different elements that go into network design to take customer requirements from an idea to actual network realization. Network design must balance elements such as: cabling, security protocols, hardware selection,

operation system selection.

Network realization is the process of transforming the logical network design into a physical topology capable of support C2 requirements. The installation phase poses its own unique challenge, taking a logical network design from abstract design diagrams to a physical network capable of supporting network operations. This process involves the network operators interpreting the network designers' plan, and manually configuring network devices one-by-one, and laying cable to build the network web.

The operations and maintenance phases of the network process requires several staff-hours per day for monitoring the computer network's health and reconfiguring or replacing network components that are affecting the reliability and survivability of the network. Failure in the planning and installation phase can have adverse effects on the operation of the network. If an operator misconfigures network devices, customers can experience network outages or prolonged downtime while troubleshooting is performed. Even iIf the configuration is a small error, countless hours can be wasted troubleshooting the network to find the errors. One way to mitigate this is to build automation in at the network design phase.

### 2.1.1 Design Tasks

There are a variety of design tasks that a network engineer must accomplish when creating the logical network diagram for a customer. The design tasks range across the spectrum from selecting cabling mechanisms, creating an Internet Protocol scheme, selecting security protocols, and designing network segmentation, to name a few. During the design process the network engineer uses the customer requirements as a guide to develop a network topology capable of meeting the customers' objectives. To accomplish this, the network engineer manually sifts through a variety of protocols and architecture design standards, and makes recommendations to create the most comprehensive logical design to meet the customer needs. The process performed by the network engineer is not only time consuming but is prone to design faults that may affect the performance of the network or leave the network vulnerable to cyber attacks. The primary weakness is the manual nature of the process.

Two of the most daunting tasks that the network engineer must complete are network segmentation and security policy enforcement. Network segmentation allows the network engineer to create different user groups from accessing resources that are not permitted. For instance, if a company has two departments, sales and supply, a network engineer would implement policies

that deny members of the supply access to sales department resources and vice-versa. Segmentation not only improves security but can minimize broadcast traffic to improve network performance. Broadcast traffic is a network frame that is transmitted to all end hosts on a local area network segment. Address resolution protocol (ARP) is an example of broadcast traffic, where an end host on a network requests a media access control address for an IP address. Broadcast traffic will increase as the number of users increase, which necessitates the need for network segmentation.

Network segmentation is achieved through the use of VLANs. VLANs perform the same exact functions as a local area networks (LAN). Both VLANs and LANs segment broadcast domains, bandwidth domains, and aid in security policy implementation. The diagram on the left in Figure 2.1 illustrates a typical LAN, segmenting users from different functional areas. It is assumed in the same figure that each of the LAN segments reside in the same building.

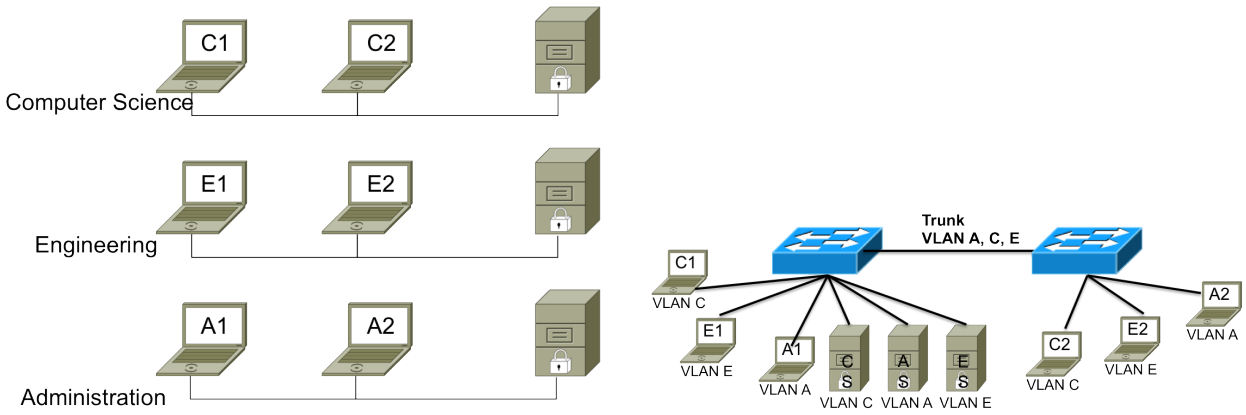


Figure 2.1: Virtual Local Area Networks

For network realization, each segment would require its own switch to ensure proper network segmentation. The need for additional network components drives up the cost of the network. It is not uncommon that users from a functional area operate in different building or across the country. VLANs can be used to help mitigate redundancy, maintain network segmentation integrity, and reduce the complexity of the VLAN design task [4]. For example, in the diagram on the right hand side of Figure 2.1, the switch on the left-hand side is in building one and the switch on the right is in building two. This setup allows multiple VLANs to operate on one switch and can be easily extended to other sites.

There are three factors to consider when designing VLANs: security policies and management

objectives; limiting broadcast domain traffic; and minimizing the total number of VLANs in a network [4]. Security policies implemented in the network determine how users are organized into to groups, based either on their status in an organization or functional work area. This organization helps control access to resources. For example, a security policy, designed to mitigate the risk of corruption, could limit the access of shipping department workers to the accounting department's resources and vice-versa. Limiting broadcast traffic is key to ensuring optimal network performance. If a single broadcast domain experiences a greater than 20 percent broadcast traffic [5], it could overwhelm the network and end user devices, thereby degrading the network's reliability. Network switches and routers have a finite amount of memory and processing power. Each VLAN that is created, generates an additional spanning tree that consumes memory and processing power. Given this, properly grouping and limiting the number of VLANs is essential to proper utilization of resources.

With an increase in cyber attacks, network security has become a high priority when designing computer networks. Figure 2.2 illustrates an example scenario that an operator might encounter.

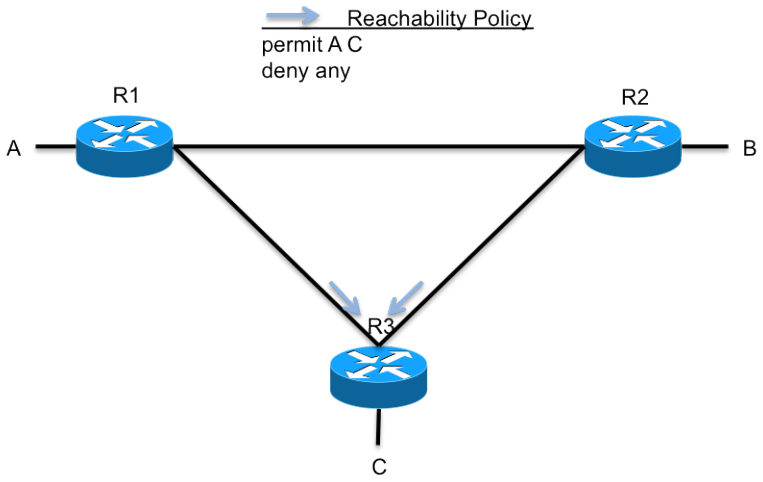


Figure 2.2: Network Access Control

A, B, C are subnets. Suppose we want to limit the traffic to subnet C. A policy might prescribe that any host in subnet A is allowed to communicate with any host in subnet C and deny all other subnets. To implement this policy, the network engineer must configure an ACL as shown in Figure 2.2 on the inbound direction of both interfaces on router R2.

There are a variety of ways that network engineers can implement security policies to achieve

reachability control. The two primary ways network engineers build reachability control into their network designs are through data or control-plane configurations. Control-plane solutions focus on denying a network router a particular route or adding black hole routes that ultimately drop the network packet according to security policies. This solution is less process intensive on network routers; however, it is not fine grained enough to block a large variety of network packets. It can only filter packets based either on the source or destination IP address. The other method implemented on the data plane is achieved through the use of packet filters, i.e., ACLs. As with the control-plane solution, ACLs are implemented based on security policies but achieve a finer granularity of control over network packets traversing the network. With more control over network traffic comes additional design decisions for the network engineer who must determined where to place ACLs based on reachability requirements, which routers the ACLs should be applied to, and which direction the traffic should be blocked, either inbound or outbound. A primary concern of the network engineer is the correctness of the ACL design. That is, any network-change event must not have an effect on the given reachability policy. Referring to Figure 2.2, a network change could involve the link between router R2 and R3 being interrupted or broken, forcing R2 to reroute traffic to R1. If the network engineer forgot to place an inbound ACL policy between router R1 and R2, then traffic from subnet B could reach subnet C, violating the reachability policy. In this scenario the reachability policy would not be consistent. Improperly placing ACLs throughout the network increases the potential to adversely affect network performance by inadvertently allowing unauthorized access to resources or inadvertently denying resource access to authorized users.

### 2.1.2 Design Approaches

Network engineers face a wide array of choices when designing the network, making their job quit demanding. To meet customer objectives, multiple design approaches are possible, some used in practice and others proposed in the literature. They can be broken down into two categories: top-down or bottom-up. Top-down network design is similar in structure to software programming or systems analysis. It first accounts for user requirements, then protocol behavior, followed by scalability requirements and technology preferences. Top-down network design allows for a flexible design that accounts for changes in either the logical or physical network. The goal of top-down network design is to ensure the network meets customer needs.

Following the tenants of top-down network design, there are two main approaches used to design the logical network defined by Cisco: the classic three-layer hierarchical design and the enterprise composite network model. The classic three-layer hierarchical design breaks up

the network design process into three distinct network layers: core, distribution and access. Each layer if designed correctly can operate autonomously without interaction with the other layers. To implement security policies and meet customers' goals it is imperative to ensure each layer is interconnected. Network engineers use the customer requirements as their right and left lateral limits when designing a network with the hierarchical method. They start by designing the access layers that provide end user network connectivity and access to network resources. During this process, the network engineers typically divide users into security groups. Once they have completed the access layer, network engineers move on to the distribution layers. The distribution layer provides the network engineer a means to implement security policies, shape traffic, and provide internal and external network connectivity to users. The distribution layer also acts as the mesh that connects the access and core layers together. Once the lower two layers are complete the network engineer then designs the core layer. This layer provides unimpeded high-speed connectivity between remote sites and the Internet. Limited security policies are applied at this level to ensure unimpeded traffic flow.

Another approach that is similar to the classic three-layer hierarchical design is the enterprise composite network model [5]. The enterprise composite network model assumes that engineers have a clear understanding of the business processes and customer requirements to properly modularize the computer network. Further, it allows the network engineer to analyze the network from a functional, logical, and physical standpoint, further breaking down the network into three main components or modules: the enterprise campus, enterprise edge, and service provider edge.

Template-based network design is another options that is heavily practiced in the military. Network engineers typically keep a historical collection of network diagrams from past operations or training exercises they participated in. These typically include LAN and wide area network (WAN) diagrams and are typically used to create standing operating procedures (SOPs) for network design and employment. SOPs, are a set of guidelines that an organization uses to define standard techniques, tactics, and procedures (TTP) for network design. Communications unit SOPs define TTPs for convoy operations, power schemes, and a variety of other tasks. Within the unit SOPs, network design TTPs are defined. Additional guidance can be found in the TRI-MEF SOP [6], which defines SOPs for network design across the Marine Expeditionary Forces (MEFs.) A network engineer uses these diagrams as starting points when designing a new network. They often take the electronic versions of the diagrams and modify as needed to achieve the operational objectives without much thought required or deviation from the template.

## 2.2 The Marine Corps Communications Planning Process

Communications planning is similar to any other form of planning in the Marine Corps. It is sequential, concurrent, repetitive, scalable, and continuous as illustrated in Figure 2.3 [7].

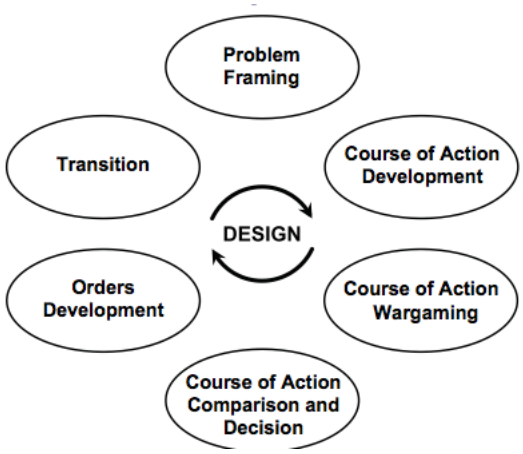


Figure 2.3: Marine Corps Planning Process. From [7]

It is also top-down, single-battle, and integrated across all staff functions. In the context of top-down planning, the ultimate responsibility of the planning process falls to the operational commander. To help achieve an effective operational plan the commander has several different staff sections that aid in the planning process. The principle staff in charge of the communication network planning is the G6/S6. The G6/S6 is involved in the initial planning process and remains highly integrated throughout the planning due to the commander's heavy reliance on communications with his force. In the problem-framing step, the G6/S6 identifies a variety of task, constraints/restraints, and assumptions from the commanders' intent and mission statement. The outputs derived from the problem-framing step are used as inputs in course of action (COA) development. During the COA development phase, a variety of COAs are created to support the commander's concept of operation. G6/S6 planners develop a plan that adheres to the principles of communications: flexible, inter-operable, reliable, survivable, timely, and secure [7]. The communications plan accounts for all communication platforms available to the unit. This ranges from single-channel radio assets to satellite communications, which provide the commander flexibility and redundancy.

The G6/S6 shops are typically divided into three main sections for planning purposes. The three sections include transmission, wire, and data. Each of these are headed by a senior officer, who

is typically a chief warrant officer and considered the subject matter expert in his assigned area. The transmission officer is responsible for planning, installation, and operation (PIO) of the radio, terrestrial, and satellite communications that provide the core communications backbone of the network. The wire officer is responsible for the PIO of the multiplexing, POTs, and physical cabling of devices. The data officer is responsible for the the development of the PIO of the packet switched network.

Communication networks are designed with a single purpose: to enable command and control for commanders. Tactical communications networks provide several vectors a commander can use to collect and synthesize information and, ultimately, disseminate orders to subordinate commanders. Technology has continued to change and provide commanders with more options to communicate across the battlefield. With the advent of the Internet, military communications have shifted from a major reliance on single-channel radios to an increased reliance on PSNs, which changed how commanders build situational awareness of the battlefield and communicate with their subordinate commanders. Commanders and troops have become highly reliant on PSNs. Commanders and troops utilize a myriad of services from email, video streaming, and other multimedia applications for is issuing orders, logistical requests, or simply communicating home. A much larger amount of data traverses the PSNs than before. Communications plays a significant role in enabling all of these tasks and mission accomplishment. To that end it is increasingly important that communications planning be detailed and thorough enough to accomplish the goal of C2l. Even though the goal of providing C2 is met, it is not done without a multitude of errors and staff hours of troubleshooting.

**Design Approach**

Of the various design approaches presented in the previous section, the Marine Corps predominately uses a template-based approach. Communications units have three main source documents that are developed during the planning process: network diagrams, cut sheets, and annex k. The network diagrams are used by the network engineers to provide a depiction of the logical network to the network operators.(Figure 1.1) Logical networks are a representation of how the network would look if all the physical devices were in the same location. The logical network does not always represent the physical topology, but it gives network operators a view of the larger network. These diagrams are living documents which can go through multiple iterations of refinement during the planning and operations phases. Cut-sheets are developed by network engineers' and provide network operators device configuration parameters to configure the network. They include IP assignment, VLAN, VTP, and ACLs, among others, that help ensure

network realization. Annex k, provides amplifying guidance to network operators. The annex k consists of broad statements that typically do not translate easily to simple instructions. These instructions direct what the network must do, not how the network is implemented.

Of the three documents developed, network diagrams and cut-sheets are used most often. The start of network planning is formed from a unit SOP, which are living documents that are routinely updated to maintain lessons learned from previous operations or training exercises. SOPs are meant to provide a starting framework for communications units in reference to all facets of communications planning. In the SOP, there is a section dedicated to network management. This section provides guidelines in an attempt to standardization network planning activities.

The network management section defines standardized VLAN assignments, root-bridge placement strategies, routing protocols, and other settings that affect network planning. The standardization found in the SOPs are derived from either personal experience of the senior network engineer, or from Defense Information Systems Agency (DISA) security technical implementation guides (STIG). DISA STIGs are set forth to ensure the best security practices are employed throughout military networks. These documents, along with best practices, are critical inputs to the design process in the Marine Corps.

**Design Workflow**

The current Marine Corps network design process is severely prone to human design faults, which leads to wasted man hours of troubleshooting and, ultimately, can lead to insecure data networks that are susceptible to cyber attacks, thus degrading the effectiveness of command and control. The typical network design pattern follows a top-down design approach, as described above, with a bottom-up refinement. Several factors affect the design decisions, including how many sites are supported, the number of users, applications on the network, and the like. The standard organization is depicted in Figure 2.4. Communications unit command structures are organized for top-down planning and loosely follow the process in Figure 2.5.

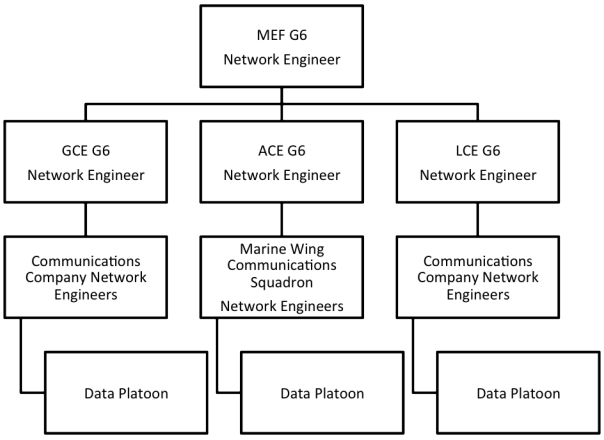


Figure 2.4: USMC Organization of Communications Planners

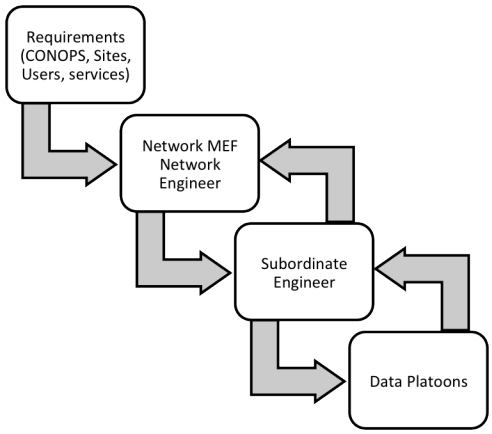


Figure 2.5: USMC Network Design Workflow

When an operation or training exercise is scheduled, the top-level network engineers begin the design process by developing an initial top-level core routing design that depicts major subordinate command connectivity. Invariably, the network engineers utilize some variant of a network diagram tool to depict the logical network design. The default choice is Microsoft VISIO. Once the core routing design is complete, the VISIO document is distributed to mid-level network engineers for validation and concurrence on the plan. In tandem, mid-level network engineers are designing internal routing and switching architectures that reside behind the core network nodes, as defined in the top-level routing design.

No matter at which level the design process occurs, there is always a back and forth dialog

and vetting of diagrams and configuration settings until, ultimately, a network design plan is created. The network engineer then verifies that the diagram meets customer requirements from their interpretation of C2 needs. Once the network design has been solidified and approved by the unit commander, the VISIO document, along with configuration guidance provided by the communications plan, is given to the network operators for installation. Figure 2.6 shows the typical work flow.

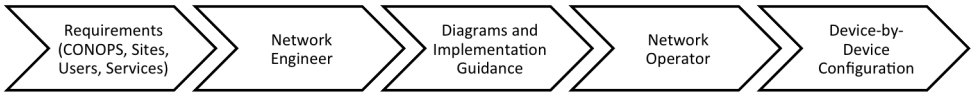


Figure 2.6: Installation Process for USMC Communications Networks

Even though the network operators are trained and certified with various levels of Cisco certifications, they typically lack experience implementing large scale networks. As a result, a majority of the work falls on the senior data operators or network engineers to ensure that the network is implemented correctly.

Further complicating the installation efforts is network operators' lack of visibility over the entire network. Network design installation is typically divided between multiple network operators who are only responsible for configuring a particular subset of devices in the overall network design. This type of process leads to tunnel vision for network operators, since they are typically only focused on their assigned tasks without regard to the overall network architecture. Once one group of operators have completed their tasks, they are usually unconcerned with the rest of the overall architecture and tend to neglect and not account for how their portion of the network implementation may affect the network as a whole. This pattern of installation will typically work for smaller size networks, such as training exercises. However, it is not realistic for large scale networks, including tactical networks that are common in real-world scenarios.

## 2.3 Current State of the Art of Top-down Design

Several network tools exist to help make network engineering jobs easier. These tools can be classified into two groups: network monitoring and network design tools. Network monitoring tools allow network engineers to monitor existing network health and quickly identify any symptomatic problems that might degrade network performance. Network design tools, on the other hand, allow a network engineer to draw a logical network architecture that will eventually be implemented by network operators. One of the most popular design tools currently available is Microsoft VISIO, which provides network engineers an array of network icons and a drawing canvas to design a logical network to meet customer specifications. While Microsoft VISIO is a good tool to use for logical network design, it is limited in that once the network engineers have completed the design, they must hand the design to network operators who will then manually configure the network devices per the guidance. Currently no application exists that allows a network engineer to create logical network architecture and then populate pre-generated configuration files for each physical device in the network.

A software solution that meets this deficiency must be able to integrate into the current processes of Marine Corps network planning. It also must ensure correctness in the design algorithms and minimize the amount of manual work required by network operators to minimize design flaws and errors. Krothapalli *et al.* conducted research that is similar to the focus of this thesis [8]. They developed a toolkit for automating and visualizing VLAN design, which employs a set of algorithms designed to assist network operators in optimizing their VLAN usage. The algorithms are a precursor to the systematic approach developed by Sung *et al.* [4]. They employ the set of algorithms on their web application, Virtual LAN Management tool. This tool allows a network operator to upload existing network configuration files into the application, which can then be used to analyze VLANs. Their tool can create a new VLAN, extend an existing VLAN, or provide a graphical depiction of the VLAN span. The user is then presented with a visual representation of required changes for the existing network to implement the new VLAN configurations. However, the tool presented is reliant on the network operators providing current network configuration files and does not focus on clean-slate network design, which is the goal of this thesis.

Prior work presented by Sung *et al.* [4], focuses on a systematic approach to network design and builds on Krothapalli *et al* [8]. The primary contribution of this work is a set of algorithms that automate two design tasks: VLAN design and ACL placement. Sung *et al.* validate their set of

algorithms on an existing network infrastructure [4]. They showed that for VLAN design they were able to reduce the overall size of VLANs. The size of a VLAN has a directed impact on the amount of broadcast traffic generated for a given VLAN. They evaluated broadcast traffic on two types of links, core and non-core. Core-links are those between core routers and links that connect to a core router. All other links are considered non-core links. By efficiently grouping hosts to minimize broadcast traffic, they were able to reduce the maximum amount of broadcast traffic by around 1000 pkts/sec and 2000 pkt/sec for non-core links and core-links, respectively [4]. For router and bridge placement they were able to reduce the average hop count by 1-1.5 hops using the systematic placement.

In contrast to the previous work, this thesis will focus on applying the algorithms presented in both [4] and [8] to clean-slate network design, that is, taking a network designer's logical network diagram and creating network-devices configuration files that are ready to be installed on all devices.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 3:
# Network Design Methodology

In this section we present the set of algorithms employed in the rapid network design (RND) application to accomplish VLAN design, ACL placement, and the creation of network configurations files.

## 3.1 VLAN Design

Sung *et al.* present a systematic algorithm to create VLANs for a given network topology [4]. The RND tool implements these algorithms to achieve a systematic approach to VLAN implementation. The algorithms are separated into two distinct phases. The first is grouping hosts into VLANs and the second is placement of the router and root bridge. For the purpose of this thesis we focus on the placement of the router and root bridge. Host grouping in the Marine Corps is driven by unit assignment, reducing the need to further group hosts.

### 3.1.1 Broadcast Cost Calculation

Router and root-bridge placement depends on the calculation of broadcast cost. This section presents the cost model to calculate that cost. *BroadcastCost* is the average amount of broadcast traffic associated with each VLAN based on the cost model below:

$$BroadcastCost_i = N_i \times B_i \times W_i$$

$N_i$ is the number of hosts per VLAN, $B_i$ is the average amount of broadcast traffic a host in $VLAN_i$ generates, and $W_i$ is the number of links present in the spanning tree for a given $VLAN_i$. When employing this algorithm, the RND tool assumes that each user generates a similar amount of broadcast traffic, which has little influence on the overall calculation when distinguishing between candidate switches for router and root-bridge placement. The cost model used in the RND application is presented below:

$$BroadcastCost_i = N_i \times W_i$$

The two variables used in the RND tool algorithm are the number of users and links in the spanning tree. The number of users are specified by the network designer during the creation of the network diagram in the RND application. The number of links in the spanning tree are then calculated utilizing algorithm q below:

**Input:** The inputs to the algorithm are: 1) User specified network diagram, created on the drawing canvas of the application; 2) VLANs specified for the network, which are obtained from the user input in the VLAN database of the application; and 3) number of users which is obtained when a user specifies clients in the RND application.

**Initialization:** This step is required to calculate the distance between each pair of network components specified in the users' network design.

**Algorithm 1**

```
1: Create graph G
2: L();                                          ▷ Number of links in the SPT
3: for all client_i in Topolgy do
4:     Calculate the path n_1 to CandidateRoot
5:     if client_i in VLAN_i then
6:         Add links to L;
7:     end if
8: end forreturn BroadcastCost_i
```

In algorithm 1, we first create a graph using the Dijkstra algorithm [9], with the candidate switch as the root. Then, for each client node that exists in the network topology, we determine if it is in the $VLAN_i$. If the client is in $VLAN_i$, we calculate the path from the client node to the candidate switch, excluding any links that exist between the client and the switch to which the client is directly connected. Once we have calculated the path, we store each path edge in $L$. After the algorithm has calculated the number of links in the spanning tree, $BroadcastCost_i$ is return based on the cost model presented above.

### 3.1.2 Finding the Root Bridge and Router

Any router or switch can be chosen as the spanning-tree root bridge for a given VLAN. The goal of root selection for the spanning-tree is to minimize the size of the tree as much as possible. Even though routers can server as a root bridge, in the Marine Corps, routers are typically only employed at the core layers of the network. Since the core of the network serves as the high-speed backbone [5] of the network, routers will be ignored for selection as a root for the spanning-tree. This limits the algorithm to either distribution or access-layer switches in the network for root selection.

When selecting a router for $VLAN_i$, either a router or a switch that can perform routing functions or a switch that is capable of implementing switched virtual interfaces is appropriate. In the

case of the Marine Corps, it is rare to find routers beyond the core routing layer. Therefore, when selecting a router for $VLAN_i$, routers are excluded and switches that can perform routing functions are evaluated. We assume each switch in the network topology either has the ability to route IP protocols or can implement switched virtual interfaces. To simplify the decision process, we assume that both the VLAN root and router are on the same device.

The goal of finding the root bridge and router is to minimize the overall traffic cost associated with a given VLAN. The general cost model is presented below:

$$Minimize[TrafficCost_i = DataTrafficCost_i + BroadcastCost_i]$$

Calculating the broadcast traffic is the same as previously presented. The data traffic cost is calculated by summing the interVLAN and intraVLAN traffic for each VLAN. This is accomplished through the following two cost models:

$$(1) InterVLAN_i = N_i \times T_i \times [d(V_i,R_i) + \Sigma f_{ij} \times d(R_i,R_j) + f_{i,INT} \times d(R_i,R_{INT})]$$

- $N_i$ is the total number of users per VLAN.
- $T_i$ is the average amount of VLAN traffic a client node in $VLAN_i$ generates.
- $d(V_i,R_i)$ is the average distance across all hosts in $VLAN_i$ from host $i$ to the designated VLAN router.
- $f_{ij}$ is the average amount of traffic that a host in $VLAN_i$ exchanges with a host in $VLAN_j$.
- $f_{i,INT}$ is the average amount of traffic a host exchanges with the internet.
- $d(R_i,R_j)$ is the hop count between $VLAN_i$ and $VLAN_j$ routers.
- $d(R_i,R_{INT}$ is the hop count between $VLAN_i$ routers and the Internet.

$$(2)\ IntraVLAN_i = N_i \times L_i \times 2d(V_i,Br_i)$$

- $L_i$ is the amount of traffic $user_i$ generates in $VLAN_i$
- $2d(V_i,Br_i)$ is the average hop count between a host in $VLAN_i$ and the spanning-tree root.

The RND application employs the interVLAN cost model in the following way. We assume that each client generates similar VLAN traffic, $T_i$. We use a constant 10kps based on results presented in Sung *et al.* [4]. Host traffic on a typical Marine Corps network only takes place between hosts and servers. Rarely are there host-to-host data exchanges. Given this, for all VLANs we ignore $d(R_i,R_j)$. We assume for the given network topology that data exchanges between client-to-server and client-to-internet are equal, therefore $f_{ij} = .5$ and $f_{i,INT} = .5$. The cost model used by RND to calculate $InterVLAN_i$ is presented below.

$$InterVLAN_i = N_i \times T_i \times [d(V_i, R_i) + f_{i,INT} \times d(R_i, R_{INT})]$$

Algorithm 2 is used to implement this cost model:

**Input** 1) $VLAN_i$ that is being evaluated, and 2) candidate root switch.

**Algorithm 2**

```
1: create graph g
2: computer shortest path
3: for each router connected to the Internet do
4:     get VLAN span
5:     calculate interVLAN cost
6:     if interVLANcost ≤ currentMin then
7:         set currentMin = interVLANcost
8:     end if
9: end for
10: return interVLANcost
```

In algorithm 2, we first create a graph using a weighted adjacency matrix graph. We then compute all pairs' shortest path using the Floyd-Warshalls algorithm [9] for each pair of vertexes in the topology. In steps 3 -6, we calculate the interVLAN cost based on the Internet routers.

RND employs the intraVLAN cost model in the following manner. We assume that the average amount of traffic of intraVLAN traffic a user generates is relatively equal. Therefore, we employ the below cost model to calculate intraVLAN traffic.

$$(2)\ IntraVLAN_i = N_i \times 2d(V_i, Br_i)$$

Algorithm 3 implements the above cost model to calculate $intraVLANcost$.

**Input** 1) $VLAN_i$ that is being evaluated and 2) candidate root switch.

## 3.2 ACL Placement

Sung *et al.*, presented a framework that finds the best location to place ACLs in a given network topology [4]. The main focus of ACL placement is the correctness and feasibility of the placement strategy. The correctness criterion ensures that reachability controls, as specified by the user, are followed. For example, if the users specify a rule that VLAN 100 cannot reach VLAN

**Algorithm 3**

```
1: create graph g
2: computer shortest path
3: s();                                                    ▷ VLAN span
4: for each client in VLAN_i do
5:     calculate VLAN span
6: end for
7: return intraVLANcost
```

101, this must always be true, regardless of the network topology changes. The feasibility criterion ensures that $\forall r, b(r) \leq c(r)$, where $b(r)$ is the total number of ACLs currently configured on a device, and $c(r)$ is the total number of ACLs allowed in the design. The framework provides four different placement strategies:

- **Minimum Rules Strategy** allows the designer to minimize the number of rules used.
$$\text{Minimize } \Sigma_r b(r)$$
- **Load Balancing Strategy** spreads the ACLs over multiple network devices to lower device overhead.
$$\text{Minimize } max_r b(r)$$
- **Capability Based Strategy** places the majority of the ACLs on devices that have a higher processing capacity.
$$\text{Maximize } min_r c(r) - b(r)$$
- **Security Centric Strategy** places the ACLs as close to the source nodes as possible to minimize the security risk. The goal of this strategy is to minimize the hop count $H$.
$$\text{Minimize } H$$

These four strategies all use the same heuristic, which initially tries to find an edge-cut set between host $i$ and $j$. The remaining steps are focused on determining on which router interface to place the ACL. This general strategy does not necessarily apply to the Marine Corps design process because the edge-cut set is relatively small in Marine Corps networks. The Marine Corps typically only employs routers at the core level and switches that are cable of ACL configurations in the core and distribution layer. The number of devices in the core and distribution layers that are capable of enforcing reachability policies is thus limited. RND employs a modified version of the algorithm. Instead of finding an edge-cut set, switches that reside in the distribution layer and have been selected as $VLAN_i$s router are candidates for ACL placement. This method is more in line with the security-centric strategy presented in Sung *et al.*, by placing the ACLs

close to the source node. We believe this strategy makes the most sense because all VLANs at some point traverse these switches to reach any other services or VLAN.

When determining on which device to place the ACL, RND chooses the router for $VLAN_i$ that was selected as the router and root bridge. Once the router is selected, the ACL is applied to the appropriate interface as stated in algorithm 4.

**Algorithm 4**

```
1: for each acl in ACL do
2:     get VLAN_i router_i
3:     add acl to router_i
4: end for
```

## 3.3 Additional Configurations

Once the router and root-bridge have been selected and ACLs placed on the appropriate network devices network configuration files are created. The following algorithms are used to create the configuration files.

Algorithm 5's primary function is to create configuration files for routers and switches that have routing functionality. For each network device, the algorithm collects user information provided in the RND application and input generated from the ACL and VLAN algorithms.

**Input** 1) Network devices that perform routing functions

**Algorithm 5**

```
1: for each router do
2:     write hostname
3:     for each interface do
4:         write IP
5:         write ACL
6:     end for
7:     for each StaticRoute do
8:         write route
9:     end for
```

```
10:     for each EIGRPRoute do
11:         write ASN
12:         write network
13:     end for
14:     for each ACL do
15:         write AccessList
16:     end for
17: end for
```

Algorithm 6's primary function is to create configuration files for switches that have no routing functionality. For each network device the algorithm collects the user information provided in the RND application and then generates the configuration files.

**Input** Network devices that perform switching function only

**Algorithm 6**

```
1: for each interface do
2:     write VLANAccess
3: end for
```

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 4:
# User Interface Design

This chapter provides an overview of the user interface of the Rapid Network Design (RND) application developed for this thesis including its design, implementation, and component parts. The application automates the process of creating network-device configuration files that can be easily installed on all network devices, thus lowering the risk of configuration errors. RND is a Java Swing application that was developed with the Java Platform, Standard Edition (Java SE) [10] [11] [12]. Netbeans Java SE was chosen because it provides an easy to use, integrated development environment that allows for the development and deployment of Java applications on desktop computers [13]. The primary layout manager used in the application was the MigLayout [14].

## 4.1 Application Features

### 4.1.1 Functional Description

Thirty-eight Java classes were developed for the RND application. The classes break down into the following categories as show in table 4.1.

Table 4.1: Class Breakdown

| Classes | Number |
|---|---|
| User Interface | 5 |
| Menus | 12 |
| Network Components | 7 |
| Algorithm | 14 |

The user interface classes contain all logic and code required for the main functionality of the program. Menu classes correspond to the configuration menus used to collect device, ACL, and VLAN configuration parameters. Network component classes are graphical depictions of network devices commonly found in a logical network diagram. These classes are employed by users to create a logical network diagram. The algorithm classes implement the algorithms found in chapter 4.

### 4.1.2 Main Client Screen

The main client screen (Figure 4.1) presents the user with a modularized view of all aspects related to designing a network with the RND application.

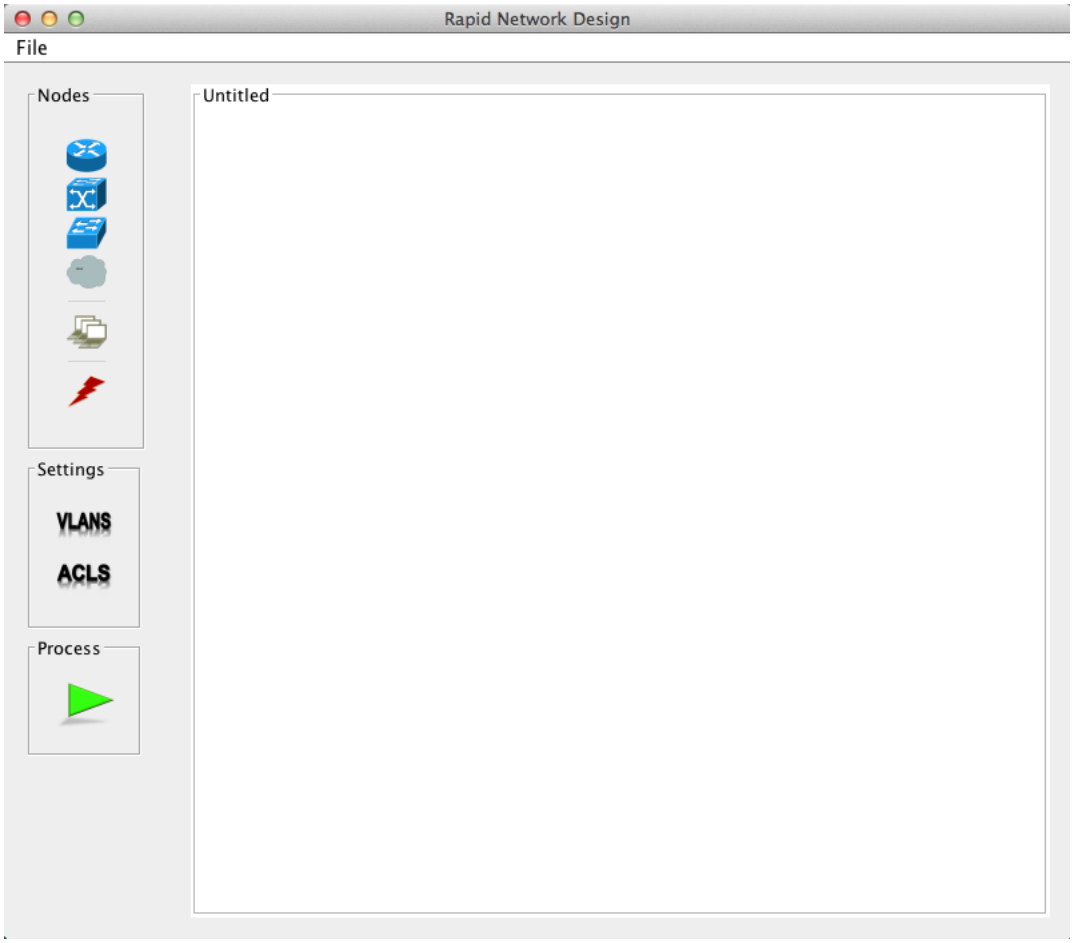


Figure 4.1: Main Interface

The canvas area is used to depict the network diagram with components from the nodes panel. Users can select any of the components from the node panel and then click on any location within the canvas area to place the component. Once users have finished adding the desired amount of components they then deselect the component they initially selected and can now add additional components. Each network component can be re-positioned by the users by clicking and dragging the component with the mouse. When users have added at least two components, they can select the connector option from the nodes panel. Connections are made between two components by sequentially selecting the two components to be linked, enabling a line to be automatically drawn between the components. Once users have completed the network design, as shown in Figure 4.2, they can then select the process configuration button.

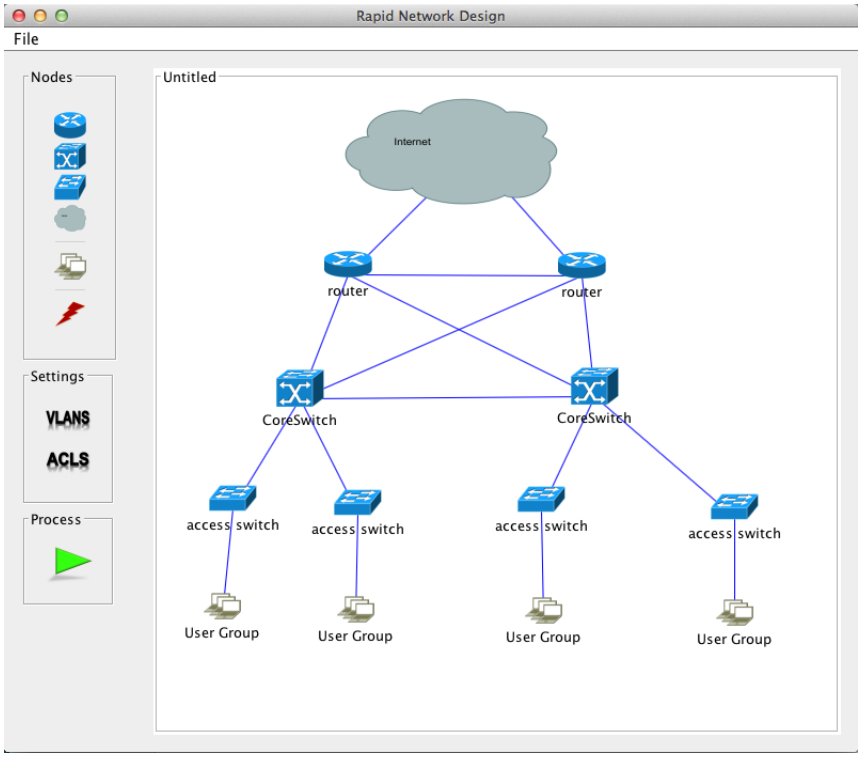


Figure 4.2: Network Design Example

Once this button is selected, the user-supplied network diagram is evaluated by the algorithms and configuration files are automatically generated for the network devices in the topology.

### 4.1.3 Nodes Panel

The nodes panel (Figure 4.3) consists of six different components: router, core switch, access switch, user group, cloud, and link connector.


Figure 4.3: Network Components

The first five components listed can be added directly to the canvas by first clicking on the Network Device and then clicking anywhere in the drawing canvas to add the component. To stop adding components, users click the stop icon in the nodes panel. Each of the network components have configuration menus that are accessed with a right mouse click on the component (Figure 4.4.)

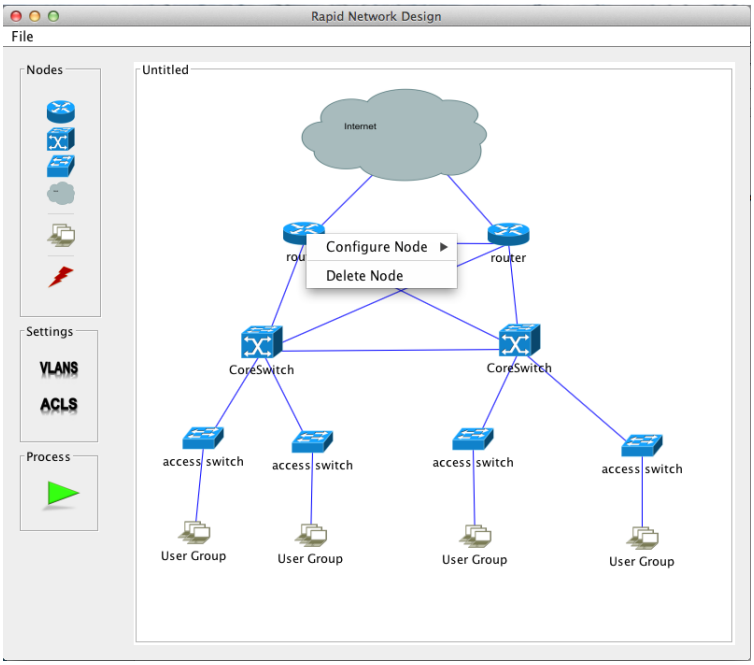


Figure 4.4: Component Configuration Selection Menu

Users have the option to either configure the node or to delete the node entirely. The component configuration menus will be described in the next section.

The sixth component, the link connector, functions differently from the network devices. When users want to connect two devices on the drawing canvas, they first select the connector icon, which is the red lighting bolt, from the nodes panel. Sequentially, they select the two devices they want connected. Once the second device has been selected a line is drawn between the two components. If users want to delete the connection, they hover the mouse over the line connecting the two devices and right-click the mouse button. They are then presented with a menu option to remove the link.

### 4.1.4 Configuration Menus

Configuration menus for the network components are depicted in Figures 4.5 to 4.8.

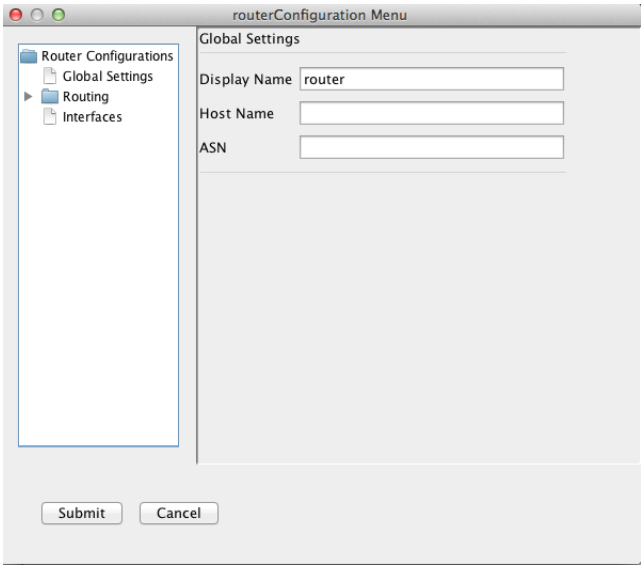


Figure 4.5: Router Configuration Menu

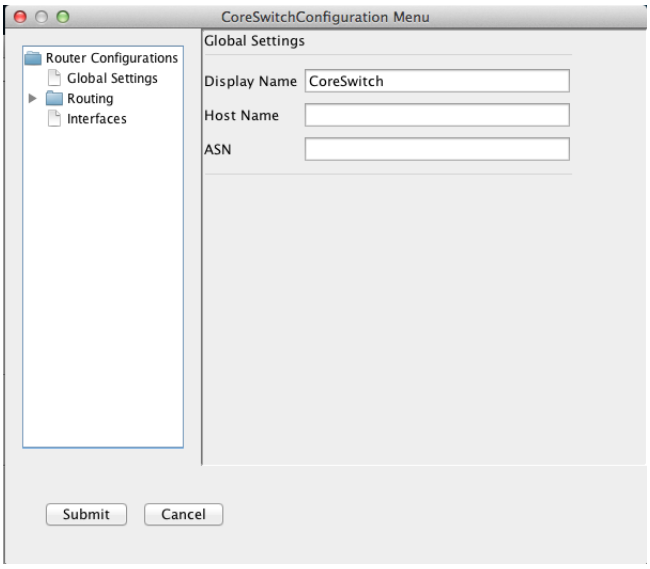


Figure 4.6: Core Switch Configuration Menu

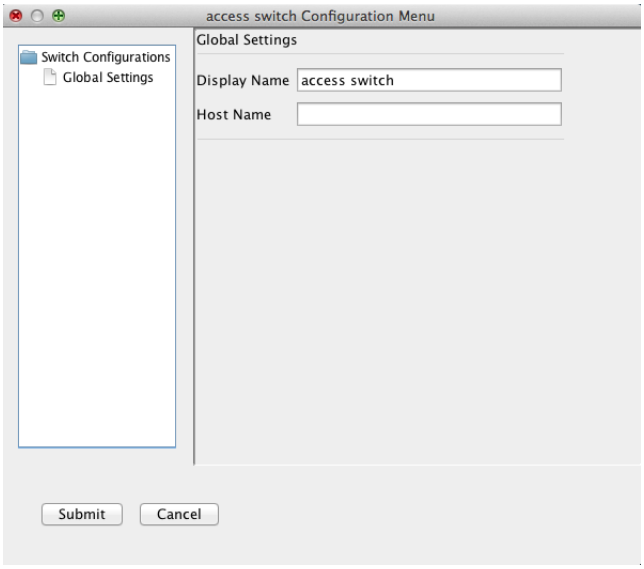


Figure 4.7: Access Switch Menu

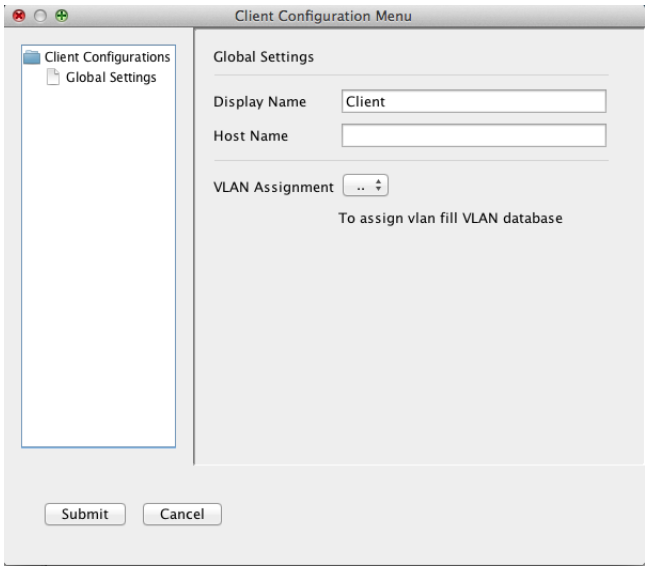


Figure 4.8: Client Configuration Menu

Each configuration menu is designed to collect user input required for network configurations, where possible combination and list boxes are used to limit input errors. For the purpose of explanation, we will focus on Figure 4.5. The remaining configuration menus function in the same manner. The configuration menus have two panes that users interact with. On the left-hand

side of the configuration menu is a tree structure that provides a simple view of the different configuration options for the network component. Each menu initially starts with the general configuration panel displayed. Users can then select a menu from the tree structure on the left to change the panel on the right-hand side and enter further information, such as routing protocols or interface IP addresses. Once users have completed configuring the network component, they must press the Submit button with the mouse to save the configurations. The configuration settings are then stored in the network device settings.

### 4.1.5 VLAN Configurations

Typically a network operator would configure VLANs directly onto a device. In the case of RND, we chose to centralize the VLANs into an arraylist structure and then place the VLANs on the devices as dictated by user group placement in the network design. Users will create VLANs by clicking the VLAN button located in the settings panel (Figure 4.9.)


Figure 4.9: VLAN Database and Process Configurations

Users are then presented with a VLAN configuration menu as shown in Figure 4.10.

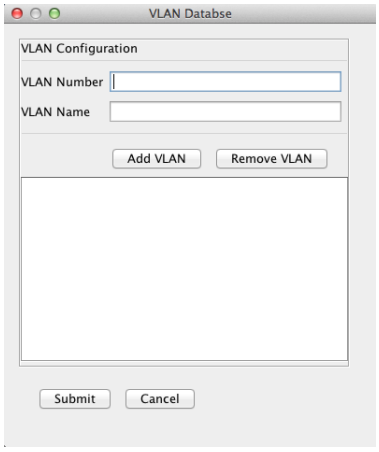


Figure 4.10: VLAN Configuration Menu

Users input the required information for a particular VLAN and then select the "add VLAN" button with the mouse. The VLAN is added to the list box. Users continues to add all VLANs for the design, after which they select the submit button to save the VLANs.

# CHAPTER 5:
# Results

In this chapter, we cover our testing method and results from automatically creating configuration files for a given network design. We take an incremental approach to testing the RND application to adequately validate each of the algorithms described in Chapter 4. First, we validate the root-bridge and router placement for a given VLAN. Next, we test the creation of the configuration files. This is followed by testing intraVLAN connectivity, interVLAN connectivity, routing protocols, and ACL placement.

## 5.1 Root-bridge and Router Placement

Validation of the correctness of the root-bridge and router placement was conducted independently from the creation of the network configuration files. We did so because the only input required is the network topology and VLANs present in the network. Two different network configurations were utilized to test the root-bridge and router placement. Figure 5.1 depicts the first network topology utilized to test the network.

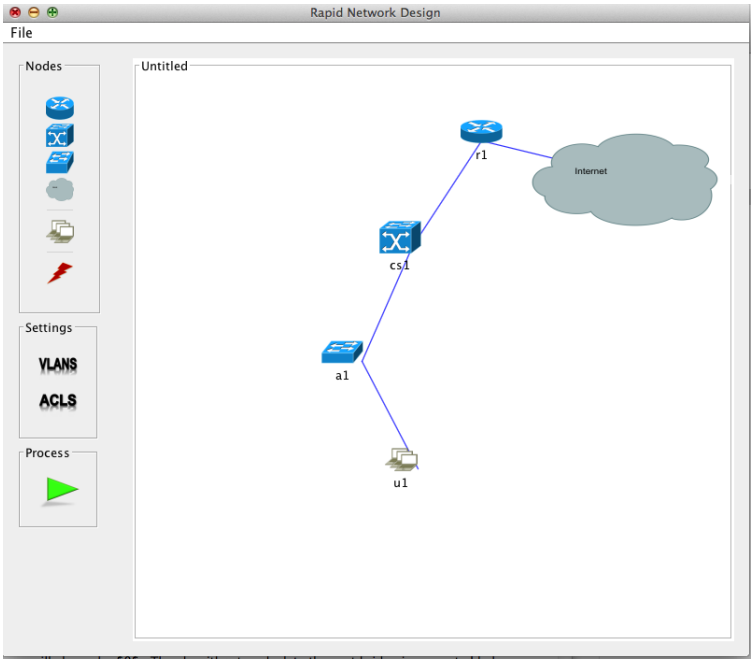


Figure 5.1: Test Network 1

The topology contains the following components listed in Table 5.1 and Table 5.2 represents the supplied parameters.

Table 5.1: Test 1 Network Components

| **Component** | **Quantity** |
|---|---|
| Router | 1 |
| Core Switch | 1 |
| Access Switch | 1 |
| Client Group | 1 |
| **Total** | **4** |

Table 5.2: Test 1 Parameters

| **VLAN** | **Number of Users** |
|---|---|
| Users | 15 |

Candidates for selection as the root-bridge and router in this topology are the CoreSwitch and the AccessSwitch. The algorithms as implemented are influenced more by interVlan traffic when computing the total cost for a given VLAN. In order to lower interVLAN traffic cost, the algorithm must decrease, $d(R_i, R_{int})$. In Figure 5.1, the distance between the internet router and the core switch is only one hop, compared to two hops for the access switch. The core switch is the more likely candidate for selection. This can also be shown by the arithmetic calculations below with the given input parameters for the test network.

- CoreSwitch:
  - $TotalCost_i = BroadcastCost_i + DataTrafficCost_i$
  - $TotalCost_i = BroadcastCost_i + InterVLAN_i + IntraVLAN_i$
  - $TotalCost_i = (N_i \times W_i) + (N_i \times T_i \times [f_{i,INT} \times d(R_i, R_{INT}]) + (N_i \times 2d(V_i, Br_i))$
  - $TotalCost_i = 15 \times 1 + 15 \times 10 \times [.5 \times 1] + 15 \times 2 \times (1 \div 15)$
  - $TotalCost_i = 15 + 75 + 2$
  - $TotalCost_i = 92$
- AccessSwitch:
  - $TotalCost_i = BroadcastCost_i + DataTrafficCost_i$
  - $TotalCost_i = BroadcastCost_i + InterVLAN_i + IntraVLAN_i$
  - $TotalCost_i = (N_i \times W_i) + (N_i \times T_i \times [f_{i,INT} \times d(R_i, R_{INT}]) + (N_i \times 2d(V_i, Br_i))$
  - $TotalCost_i = 15 \times 0 + 15 \times 10 \times [.5 \times 2] + 15 \times 2 \times (0 \div 15)$
  - $TotalCost_i = 0 + 150 + 0$
  - $TotalCost_i = 150$

The results from running the RND application are shown in Figure 5.2. The RND application produced the results expected from hand verification.

```
Calculating rootBridge for Vlan: users
-------------------------------------------------
Candidate Switch is: cs1
BroadcastCost: 15.0
InterVlanCost: 75.0
IntraVlanCost: 2.0
Total Cost: 92.0
*************************************************
Candidate Switch is: a1
BroadcastCost: 0.0
InterVlanCost: 150.0
IntraVlanCost: 0.0
Total Cost: 150.0
*************************************************
Vlan: users
The root is: cs1
The router is: cs1
*************************************************
```

Figure 5.2: Test 1 Network Results

The next test performed validated a VLAN that spans over multiple network devices, as well as adding additional VLANs to the network. The second test network contains the following components listed in Table 5.3 and Table 5.4 represents the input parameters.

Table 5.3: Test 2 Network Components

| **Component** | **Quantity** |
|---|---|
| Router | 2 |
| Core Switch | 3 |
| Access Switch | 4 |
| Client Group | 5 |
| **Total** | **14** |

Table 5.4: Test 2 Parameters

| **VLAN** | **Number of Users** |
|---|---|
| Users | 15 |
| Servers | 5 |
| Admin | 5 |

Figure 5.3 is the network topology used for this test.

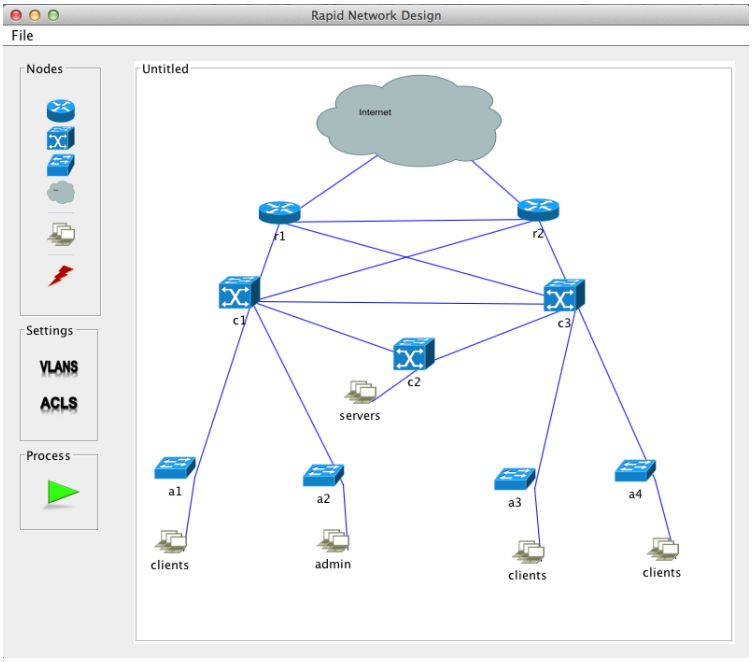


Figure 5.3: Test 2 Network Topology

In this topology, for each VLAN there are seven candidate devices for the root-bridge and router. Figure 5.4 shows the results for each VLAN.

```
Vlan: users
The root is: c3
The router is: c3
**************************
Vlan: admin
The root is: c1
The router is: c1
**************************
Vlan: servers
The root is: c1
The router is: c1
**************************
```

Figure 5.4: Test 2 Results

Calculated results for each VLAN produced by the RND application are shown in Figures 5.5 and 5.6.

```
Calculating rootBridge for Vlan: users
-------------------------------------------------------
Candidate Switch is: c1
BroadcastCost: 60.0
InterVlanCost: 75.0
IntraVlanCost: 10.0
Total Cost: 145.0
*******************************************************
Candidate Switch is: c2
BroadcastCost: 75.0
InterVlanCost: 150.0
IntraVlanCost: 12.0
Total Cost: 237.0
*******************************************************
Candidate Switch is: c3
BroadcastCost: 60.0
InterVlanCost: 75.0
IntraVlanCost: 8.0
Total Cost: 143.0
*******************************************************
Candidate Switch is: a1
BroadcastCost: 60.0
InterVlanCost: 150.0
IntraVlanCost: 12.0
Total Cost: 222.0
*******************************************************
Candidate Switch is: a2
BroadcastCost: 75.0
InterVlanCost: 150.0
IntraVlanCost: 16.0
Total Cost: 241.0
*******************************************************
Candidate Switch is: a3
BroadcastCost: 60.0
InterVlanCost: 150.0
IntraVlanCost: 10.0
Total Cost: 220.0
*******************************************************
Candidate Switch is: a4
BroadcastCost: 60.0
InterVlanCost: 150.0
IntraVlanCost: 10.0
Total Cost: 220.0
*******************************************************
```

```
Calculating rootBridge for Vlan: admin
-------------------------------------------------------
Candidate Switch is: c1
BroadcastCost: 5.0
InterVlanCost: 25.0
IntraVlanCost: 2.0
Total Cost: 32.0
*******************************************************
Candidate Switch is: c2
BroadcastCost: 10.0
InterVlanCost: 50.0
IntraVlanCost: 4.0
Total Cost: 64.0
*******************************************************
Candidate Switch is: c3
BroadcastCost: 10.0
InterVlanCost: 25.0
IntraVlanCost: 4.0
Total Cost: 39.0
*******************************************************
Candidate Switch is: a1
BroadcastCost: 10.0
InterVlanCost: 50.0
IntraVlanCost: 4.0
Total Cost: 64.0
*******************************************************
Candidate Switch is: a2
BroadcastCost: 0.0
InterVlanCost: 50.0
IntraVlanCost: 0.0
Total Cost: 50.0
*******************************************************
Candidate Switch is: a3
BroadcastCost: 15.0
InterVlanCost: 50.0
IntraVlanCost: 6.0
Total Cost: 71.0
*******************************************************
Candidate Switch is: a4
BroadcastCost: 15.0
InterVlanCost: 50.0
IntraVlanCost: 6.0
Total Cost: 71.0
*******************************************************
```

Figure 5.5: Test 2 VLAN Results

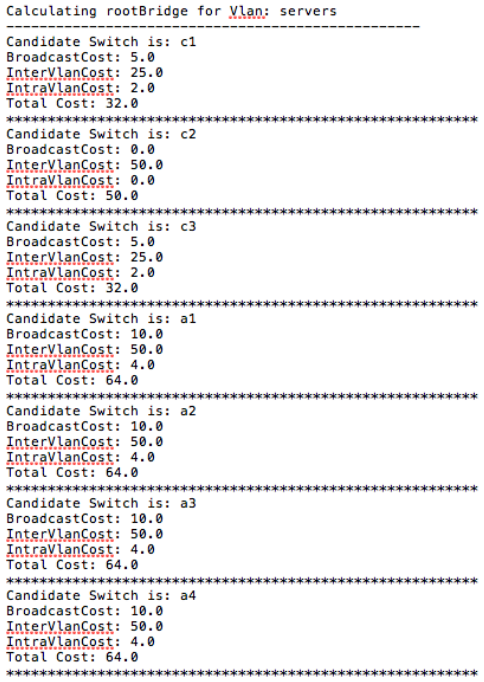


Figure 5.6: Test 2 VLAN Server Results

VLAN admin and servers chose the root in a similar fashion to the initial test presented above. For the user VLAN, C3 is chosen as the root-bridge and router. Both C1 and C3 in the topology have the same broadcast traffic and interVLAN traffic. Therefore, the distinction is between the intraVLAN traffic. Since $d(V_i, Br_i)$ is smaller in the case of C3, as shown below, this confirms the appropriate switch was selected.

- C1:
  - $IntraVLAN_i = N_i \times 2d(V_i, Br_i)$
  - $intraVLAN_i = 15 \times 2((1+2+2)/15)$
  - $intraVLAN_i = 10$
- C2:
  - $IntraVLAN_i = N_i \times 2d(V_i, Br_i)$
  - $intraVLAN_i = 15 \times 2((2+1+1)/15)$
  - $intraVLAN_i = 8$

## 5.2 Network Configuration Validations

Figure 5.7 is the test network that was used to validate the algorithms presented in Chapter 4.

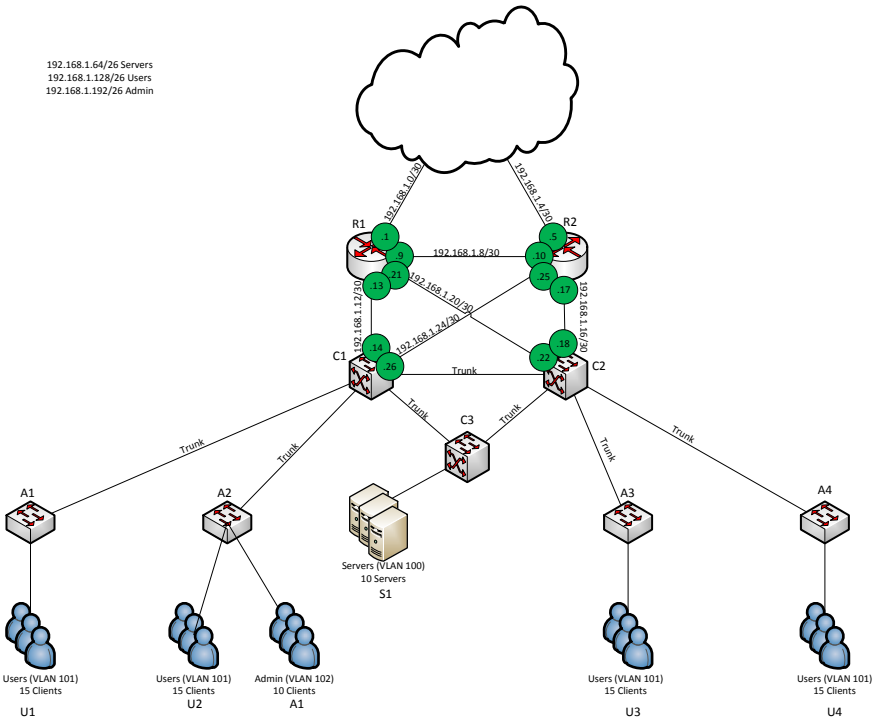


Figure 5.7: Test Network Diagram

The network consists of a total of nine network devices. The two routers and two core switches are each configured with four interfaces as depicted on the network diagram. Any connection between a core switch and access switch is assumed to be a trunk link and is configured accordingly. When the RND application runs, it will produce nine separate configuration files for each network device shown in the network diagram. Internal routing protocols are configured on the router and core switches along with all IP interfaces. All switch devices are configure with each VLAN that was supplied by the users. The access switch ports are configure for the appropriate number of user and VLAN access as designated in the network diagram. After the RND application has created the configuration files, the files are validated in a virtual environment using Packet Tracer and a physical test lab.

Each setup utilized the same network topology and input parameters and testing tools such as: command line utilities, ping and traceroute, for validation. The main difference between the two testing environments is capturing network traffic to validate network settings. For example,

the test run on Packet Tracer is easily validated using the simulation mode. For the physical setup, we utilized tcpdump to capture the network traffic. The next section describes both the virtual and physical network setups.

### 5.2.1 Environment Setup

**Packet Tracer**

For the virtual environment the set up utilized the following components:

- Host machine: MAC Book Pro OSX version 10.8.4
- VMWare Fusion version 5.0.3
- Windows 7 Profession service pack 1
- Packet Tracer version 5.3.1.004

VMWare Fusion was installed on the host machine. Then a virtual machine was created using the Windows 7 operating system. Packet Tracer was installed onto the Windows 7 VM. Once all software was validated as running correctly, we created an empty project file in Packet Tracer. Based on Figure 5.7, the device selection was as follows:

Table 5.5: Packet Tracer Network Devices

| Device | Quantity | GE ports | FE Ports | E Ports | Operation System |
|---|---|---|---|---|---|
| C2811 | 2 | 0 | 2 | 4 | Cisco 2800 version 12.4(15)T1 |
| C3560 | 3 | 2 | 24 | 0 | Cisco 3560 version 12.2(37)SE1 |
| C2960 | 4 | 2 | 24 | 0 | Cisco 2960 version 12.2(25)FX1 |

**Physical Lab**

The physical test lab setup utilized the components listed below and network devices listed in Table 5.6.

- MAC Book Pro OSX version 10.8.4
- Toshiba Satellite Windows 7 service pack 1

Table 5.6: Test Lab Network Devices

| Device | Quantity | GE ports | FE Ports | E Ports | Operation System |
|---|---|---|---|---|---|
| Catalyst 2960 | 2 | 2 | 24 | 0 | Cisco 2960 version 12.2(25)SEE2 |
| Cisco 2800 | 3 | 2 | 24 | 0 | Cisco 2800 version 12.4(3i) |
| Cisco 2600 | 2 | 0 | 2 | 4 | Cisco 2600 version 12.2(7) |

Of note, when we conducted the testing scenario with the physical lab setup we encountered configuration challenges induced by the Cisco operating system's specific commands. The

operating system running on the Cisco device can have a large impact on which configuration commands are supported. In addition, this can affect the order in which commands can be executed. The main issues we encountered involved the Cisco 2800 operating system. The Cisco 2800 device we used to represent a core switch is an integrate service router with a broad range of interface options [15]. The device we used for the testing scenario was equipment with two gigabitethernet ports and eight fastethernet ports.

When we ran our initial tests in the virtual environment the switches utilized the command pattern listed in Table 5.7 [16].

Table 5.7: Test Lab VLAN Commands

| | **Command** | **Purpose** |
|---|---|---|
| Step 1 | enable | Enables privileged EXEC mode |
| Step 2 | conf t | Enter global configuration mode |
| Step 3 | vlan *number* | Specifies the VLAN to configure |
| Step 4 | name *name* | Names the VLAN |

Our initial algorithm which writes configuration files only accounted for this initial case. We attempted to load the configuration file on the Cisco 2800 router and received several errors. The primary error was the VLANs were properly loaded on the router. After researching the command database for the Cisco 2800, we discovered that the sequence of steps to enter VLANs on the router was different from our initial test equipment in the virtual environment. We had to add additional code in our algorithm to write the configuration commands in the appropriate order given that device was Cisco 2800. The command sequence is listed below in 5.8.

Table 5.8: Cisco 2800 VLAN Commands

| | **Command** | **Purpose** |
|---|---|---|
| Step 1 | enable | Enables privileged EXEC mode |
| Step 2 | vlan database | Enters the switch module VLAN database |
| Step 3 | vlan *number* name *name* | Names the vlan |
| Step 4 | exit | Exits the VLAN configuration and saves the database |

After performing these steps all other commands execute as expected.

**Additional Configuration Parameters**

Once the network was established the diagram was replicated in the RND application using the devices listed in the virtual and physical setup and the following parameters:

- VLANs
  - users : 20

  - admin : 5
  - servers : 5
- ACLs
  - deny users -> admin
  - deny servers -> admin
- Internal Routing Protocol
  - EIGRP
    - network 192.168.1.0

Interfaces are configured based on the diagram IPs and trunk links. Once all information is entered into the RND application, the network topology is evaluated by the algorithms. Network device configuration files are created and save based on the host name specified by the user. For the test network, the host names match the names provided on the diagram. Prior to loading the configuration files onto a network device, a visual inspection of the files is conducted to ensure formatting is correct and the output seems logical. Also, we inspected that the root-bridge and router assignment for each VLAN was assign appropriately. After the files are created we upload the files to the network devices in both test environments.

**IntraVlan Communication Validation**

After the configuration files were uploaded onto each device in Packet Tracer, we chose to validate interVLAN communication first. We chose interVLAN communication fist, because as long as the switches participating in the VLAN are configured properly they will be unaffected by any mis-configurations in the routing architecture.

We configure six laptops on the network as depicted in Figure 5.7. We assigned each of the laptops an IP address from the appropriate VLAN and then conducted testing. From users node 1, connected to access switch a1, we issued the ping command to user node 4. The results of the ping command are in Figure 5.8.

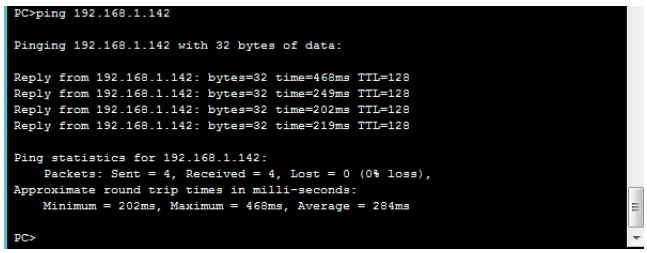


Figure 5.8: Ping Test u1 to a1 Virtual Test Lab

To further validate the this result we placed Packet Tracer in simulation mode and issued the same command and followed the path of the ICMP packets. We set the filters to view only ICMP traffic. Observing the ICMP traffic, we identified that the traffic only traversed trunk links and did not traverse any links configured as routing links.

For the physical test lab, we setup an additional monitoring laptop on core switch c1. We then configure the switch to monitor traffic using the switch port analyzer setup [17] to capture the traffic.

**InterVlan Communication Validation**

To test interVLAN traffic we chose to test users-to-servers and admin-to-users. These two groups were chosen because of ACLs configured, which limits access between the user and admin VLANs. ACL validation will be discussed later. First we tested interVLAN routing between the clients and the servers. First, we issued a ping command from u1 to s1. The results of the ping are depicted in Figure 5.9.

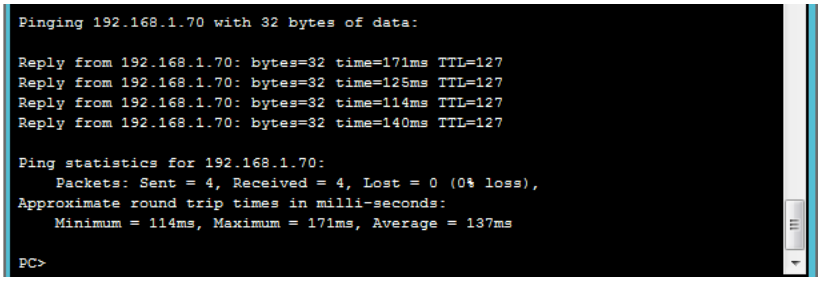


Figure 5.9: Ping Test u1 to s1

After the successful ping from u1 to s1, we tested connectivity between the admin and user VLAN. We utilized traceroute to verify connectivity. Due to ACL configurations, the ping command will fail because users are not allowed to communicate with the admin VLAN. The result of the traceroute command (Figure 5.10) show that the ICMP echo and echo-reply's fail, but eventually the traceroute completes and shows that it made it to the end destination of u1.

```
PC>tracert 192.168.1.140

Tracing route to 192.168.1.140 over a maximum of 30 hops:

  1   62 ms     78 ms     46 ms     192.168.1.193
  2   *         *         *         Request timed out.
  3   *         *         *         Request timed out.
  4   *         *         *         Request timed out.
  5   *         *         *         Request timed out.
  6   52 ms     98 ms     125 ms    192.168.1.140

Trace complete.

PC>tracert 192.168.1.140
```

Figure 5.10: Ping Test a1 to u1

**Routing Validation**

There are two main elements we are concerned with when validating routing configurations. First is interface configurations. This ensures that the correct IP address and network mask is configured properly. This can either be achieved by manual inspection of the configuration file prior to loading it on the network device or by issuing the "show run" command at the device command prompt after the configuration file is loaded. For each device that required an IP address, we manually inspected the configuration files prior to loading them onto the network devices. Figure 5.11 shows an example of a correct configuration of an interface.

```
interface FastEthernet0/0
 ip address 192.168.1.6 255.255.255.252
 duplex auto
 speed auto
!
```

Figure 5.11: Interface Configurations

The next part of the routing protocol to inspect is the internal routing. First we used manual inspection to validate EIGRP configurations were correct. Figure 5.12 shows an example of a correct EIGRP configuration.

```
router eigrp 3000
 network 192.168.1.0
 auto-summary
!
```

Figure 5.12: EIGRPConfig

After confirming the configurations correctness, we loaded the configuration files onto their respective network devices. We then issued the "show IP route" command at the device command

prompt, to ensure each device was properly learning EIGRP routes. Figure 5.13 shows an example of a routers routing table after the "show ip route" command has been issued. In this example the router has three directly connected routes, that it learned from configurations and five additional routes that were learned by EIGRP.

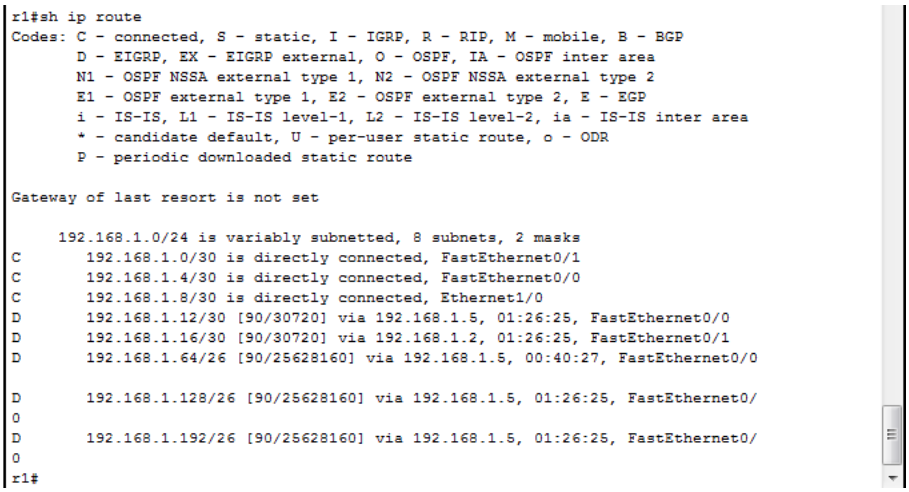


Figure 5.13: EIGRP Route Discovery

### ACL Validation

The last element of the network we inspected was ACL configuration. Two rules were configured for the network:

- permit users to access the server VLAN
- deny users access to the admin VLAN

First we visually inspected the configuration file for proper ACL writing. Then we checked that the rule worked as expect by using the ping utility. First we issued a ping command from u1 to a1. Figure 5.14 shows the ping command failed as expected. Next we tested that users could access the server VLAN. Figure 5.15 shows the ping command succeeded as expected.

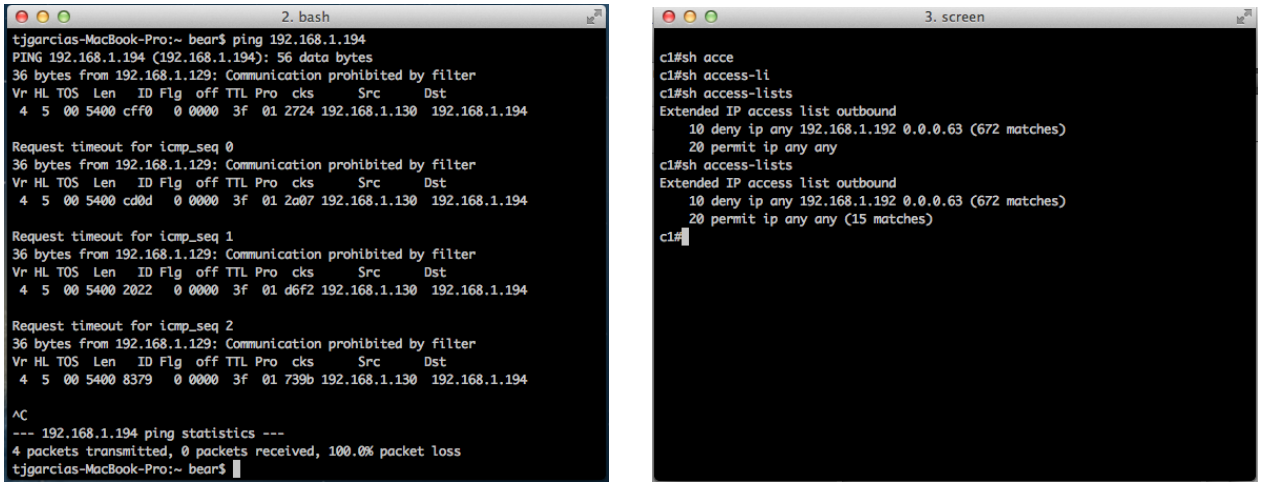


Figure 5.14: ACL Validation

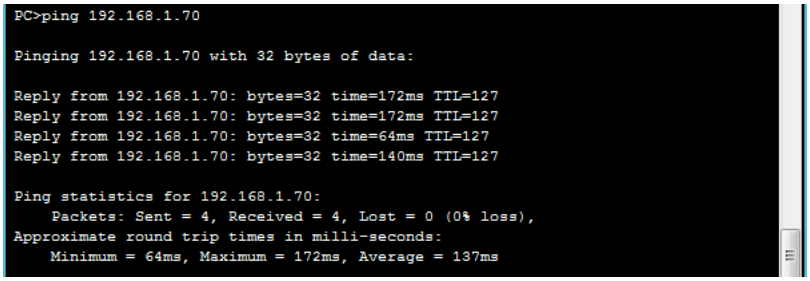


Figure 5.15: ACL Validation

The last element to inspect for ACLs, is the placement of the ACL. The algorithm presented in 3.2, places the ACL on the interface nearest the source of the traffic. To ensure this is the case for our ACL placement, we visually inspected the configuration files. After visual inspection we are confident the placement is correct.

# CHAPTER 6:
# Conclusions and Future Work

This section discusses the overall results of our research and offers suggestions on areas of future work to advance the RND application to support a wider range of network designs.

## 6.1 Conclusion

The goal of our research was to investigate the feasibility of developing a software application that can automatically create network-device configuration files from user input of a physical topology and a set of high-level design objectives. We were most interested in eliminating inefficiencies and misconfiguration errors in the Marine Corps network design process.

Our approach in developing an automated solution involved developing of a user interface to provide similar capabilities to existing software utilized by Marine Corps network engineers. RND leverages existing systematic design algorithms to find the optimal router and root-bridge placement for a VLAN and determine the rules and placement of ACLs. We were successful in automatically creating configuration files that could be easily uploaded to network devices without any customization. This confirms the potential of systematic network design approaches in eliminating manual configurations practices.

We have successfully demonstrated the ability to automatically create network device configuration files for a Marine Corps branch network with a dozen of routers and switches. Our hope is that this work serves as the first step toward the deployment of an automated solution for network design in the Marine Corps, thus creating efficiencies, lowering costs and communications failures.

## 6.2 Application Limitations

Due to time constraints and the scope of this thesis, some features are not supported by the software:

- Where possible, we pre-populated combo boxes and list boxes on the GUI. This was not possible for some input, which requires a user to manually enter the information.
- Input validation in general is not currently implemented. If a users inputs erroneous data in inputs fields, there is a potential to crash the application. In its current state, users must

input network parameters in a correct and consistent fashion with standard practices.

- IP validation: If an incorrect IP is entered into any of the device configuration menus, ACL menu, or VLAN menu, the input is accepted. The input is then written into the configuration file verbatim. If an incorrect IP is entered, this will cause a mis-configuration in the device configuration file.
- Support full-featured ACLs. The primary focus of this work was the correct placement of an ACL versus writing intricate ACLs. RND only provides the capability to deny or permit traffic between whole VLAN subnets and does not provide a fine grain strategy for ACL writing.
- Visual depiction of multiple connections. RND provides the ability to configure multiple links on a device; however, the capability to visually depict multiple links between network devices on the drawing canvas is not supported.

## 6.3 Future Work

The prototype design presented in this thesis provides a foundation on which to build a fully operational application that can be further refined, resulting in a working prototype that can be tested and evaluated by organizations interested in such an application, such as MARCORSYSCOM and C4I. The following are recommendations for future work in the development of the rapid network design application:

- Conduct a real-world usability test of a working prototype with Marine Corps network engineers. Incorporate feedback and recommendations. This step is vital to incorporating a new network design philosophy into the Marine Corps.
- Incorporate a centralized equipment database for all network equipment that contains current module interfaces installed. Cisco has strict naming standards when configuring network device interfaces and the naming convention can change between devices. Currently, RND utilizes hard coded device parameters to demonstrate the capability to automate network design. Incorporation of a centralized equipment database will allow for a more robust application capable of supporting a variety of devices.
- Support full-featured ACLs. The primary focus of this thesis was ACL placement. Given this fact, we did not fully focus on the multiple ways how an ACL can be written and determine the most effective way in which to write these rules. Incorporation of a more robust ACL writing algorithm will improve the security.
- Provide a capability to automatically upload generated configuration files to network de-

vices. RND, as implemented, does not have the ability to automatically upload device configuration files to network devices. Incorporation of this capability would reduce the risk of uploading the incorrect configuration file to a network device.

THIS PAGE INTENTIONALLY LEFT BLANK

# List of References


[1] D. Caldwell, A. Gilbert, J. Gottlieb, A. Greenberg, G. Hjalmtysson, and J. Rexford, "The cutting EDGE of IP router configuration," *ACM SIGCOMM Computer Communication Review*, vol. 34, pp. 21–26, Jan. 2004.

[2] Z. Kerravala, *"Configuration management delivers business resiliency,"* November 2002. Boston: The Yankee Group, no longer published online.

[3] Marine Wing Communications Squadron - 38, *"MWCS-38 Communications Standard Operation Procedures,"* Jun 2011.

[4] Y.-W. E. Sung, S. G. Rao, G. G. Xie, and D. A. Maltz, "Towards systematic design of enterprise networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, pp. 695–708, June 2011.

[5] P. Oppenheimer, *Top-Down Network Design*. Indianapolis, IN: Cisco Press, 2004.

[6] Tri-MEF Communications SOP Working Group, *"Tri-MEF Communications Standard Operation Procedures(SOP) version 3,"* Oct 2009.

[7] United States Marine Corps, *MAGTF Communications System 3-40.3*. Washington, DC: Army Publishing Directorate, 2010.

[8] S. D. Krothapalli, X. Sun, Y.-W. E. Sung, S. A. Yeo, and S. G. Rao, "A toolkit for automating and visualizing VLAN configuration," in *SafeConfig '09 Proceedings of the 2nd ACM Workshop on Assurable and Usable Security Configuration*, 2009.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introductions to Algorithms*. Cambridge, MA: The MIT Press, 2007.

[10] J. Marinacci and C. Adamson, *Swing Hacks*. Sebastopol, CA: O'Reilly Media, 2005.

[11] K. Sierra and B. Bates, *Head First Java (2nd ed.)*. Sebastopol, CA: O'Reilly Media, 2003.

[12] M. Loy, R. Eckstein, D. Wood, J. Elliott, and B. Cole, *Java Swing (2nd ed.)*. Sebastopol, CA: O'Reilly Media, 2003.

[13] Oracle Technical Network, "Java SE at a glance." [Online]. Available: http://www.oracle.com/technetwork/java/javase/overview/index.html

[14] MiGLayout, “MiG Layout.” [Online]. Available: http://www.miglayout.com/

[15] Cisco Systems, “Cisco 2800 Series Integrated Service Routers.” [Online]. Available: http://www.cisco.com/en/US/products/ps5854/index.html

[16] Cisco Systems, “Configuration: Basic Software Configuration Using the Cisco IOS Command-Line Interface.” [Online]. Available: Configuration: BasicSoftwareConfigurationUsingtheCiscoIOSCommand-LineInterface

[17] Cisco Systems, “Configuring SPAN and RSPAN.” [Online]. Available: http://www.cisco.com/en/US/docs/switches/lan/catalyst3550/software/release/12.1_13_ea1/configuration/guide/swspan.html

# Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia
2. Dudly Knox Library
   Naval Postgraduate School
   Monterey, California